



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/002,505	11/01/2001	Bernard Iulo	POU920010026US1	4214

7590 07/09/2004

Philmore H. Colburn II
Cantor Colburn LLP
55 Griffin Road South
Bloomfield, CT 06002

EXAMINER

SANTOS, PATRICK J D

ART UNIT	PAPER NUMBER
----------	--------------

2171

DATE MAILED: 07/09/2004

6

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

10/002,505

Applicant(s)

IULO, BERNARD

Examiner

Patrick J Santos

Art Unit

2171

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 27 April 2004.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 17-33 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 17-33 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.
2. Claims 17-18, 20-22, 24-26, and 30-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over the publication "MICROSOFT BACKOFFICE (TM) Administrator's Survival Guide," by Arthur Knowles, published by SAMS PUBLISHING (TM), 1996 (hereafter Knowles '96) in view of the press release, "Desaware (TM) Event Log Toolkit (TM)" released October 2000, and obtained from the www.desaware.com web site as archived by the WAYBACK MACHINE (www.archive.org) on Dec. 4, 2000 (hereafter Desaware '00), and the publication, "Advanced Windows (TM)" by Jeffrey Richter, published by Microsoft Press (TM), 1997 (hereafter Richter '97).

Claims 17 and 24:

Regarding Claim 17, Knowles '96 discloses the EVENT VIEWER (TM) utility in MICROSOFT WINDOWS (TM). Specifically, Knowles '96 discloses: a method, and storage medium encoded with machine-readable program code to implement said method, for managing computer performance, comprising:

- for each task executing on a computer system, performing:
 - o capturing details relating to said execution while said task is active; and

- storing said details in an activity record (Knowles '96: pp. 179-185 – note that event logging covers all task executions on a system and further that an event log record reads on an activity record); and
- upon detecting an error relating to a task:
 - retrieving activity records associated with said task in which said error was detected (Knowles '96: p. 180 – see sidebar Note discussing detecting system errors); and
 - analyzing said activity records to determine a root cause of said error (Knowles '96: pp. 181-182 – note section titled, “Viewing Events”);
 - wherein said details in an activity record include:
 - a task name assigned by a user of said computer system (Knowles '96: p.182, Table 6.4 – note the Source item is defined as the application name i.e. task name); and
 - a flag (Knowles '96: p.182, Table 6.4 – note the Event ID item reads on a flag).

However, the EVENT VIEWER (TM) of Knowles '96 does not explicitly disclose that the flag specifies whether said task was successfully completed.

Desaware '00 discloses the EVENT LOG TOOLKIT (TM) that enables making custom events i.e. custom flags, for applications that are visible in the EVENT VIEWER (TM) of Knowles '96 (Desaware '00: “Desaware’s Event Log Toolkit (TM) for Windows 2000 (TM) and Windows NT 4.0 (TM)”, pp. 1-2). However, Desaware '00 does not explicitly disclose the flag is an indication whether or not the task was successfully completed.

Richter '97 discloses the GetExitCodeProcess() function, which is part of the MICROSOFT WIN32 API (TM). Specifically, the GetExitCodeProcess() function of Richter '97 enables sending a status to the operating system whether or not a task was successfully completed (Richter '97: p. 69, ln. 19).

It would have been obvious to a person having ordinary skill in the art to apply the toolkit of Desaware '00 to the EVENT VIEWER (TM) of Knowles '96. The motivation to combine is suggested by Desaware '00 in that the toolkit provides an easy-to-use means to create events specifically targeting the EVENT VIEWER (TM) of Knowles '96 (Desaware '00: "Desaware's Event Log Toolkit (TM) for Windows 2000 (TM) and Windows NT 4.0 (TM)", p. 1, paras. 1-3).

It would have been further obvious to a person having ordinary skill in the art to apply the exit code notification means of Richter '97 to the Knowles '96 and Desaware '00 combination. The motivation to combine is suggested by Richter '97 that notification that a task has not successfully completed provides debugging information for dependent tasks (e.g. child processes) (Richter '97: p. 69, lns. 19-25).

Claims 18, 20, and 25:

Regarding Claims 18, 20, and 25, Knowles '96, Desaware '00, and Richter '97 in combination disclose all the limitations of Claim 17 (supra). Additionally, Knowles '96, Desaware '00, and Richter '97 in combination disclose:

- (Claims 18 and 25)
 - o for said each task executing on a computer system, performing (Knowles '96: pp. 179-185):

- capturing environmental information associated with computer system activity that occurs while said task is active;
 - creating an environmental record; and
 - storing said environmental information in said environmental record (Richter '97: pp. 43-47, section titled, "A Process's Environmental Variables – note that this environmental information stored in an EVENT VIEWER (TM) record reads on an environmental record); and
- if said analyzing said activity records does not reveal a root cause of said error:
- retrieving environmental records created between a last successful execution of a program associated with said task and a time in which said error occurred (Knowles '96: p. 180 – see sidebar Note discussing detecting system errors); and
 - analyzing said environmental records in conjunction with said activity records to determine a root cause of said error (Knowles '96: pp. 181-182 – note section titled, "Viewing Events");
 - wherein said environmental information includes:
 - a name of said program (Knowles '96: p.182, Table 6.4 – note the Source item is defined as the application name i.e. task name); and
 - environmental flags operable for classifying said task (Richter '97: pp. 43-47, section titled, "A Process's Environmental Variables").

Art Unit: 2171

- (Claim 20) wherein said capturing details relating to said execution further includes capturing (Knowles '96: p. 182, Table 6.4):
 - o computer system time stamp at task start;
 - o computer system time stamp at task end;
 - o an activity record identifier;
 - o amount of initial storage allocated to said task;
 - o name of an initial program invoked for said task;
 - o a pointer to a list of additional storage requests for said task;
 - o a count of a number of additional storage requests made for said task;
 - o a pointer to a list of data sets or file names used by said program;
 - o a pointer to a list of locks and latches used by said program; and
 - o a count of a number of locks and latches used by said program.

Claim 21:

Regarding Claim 21, Knowles '96, Desaware '00, and Richter '97 in combination disclose all the limitations of Claim 18 (supra). Additionally, Knowles '96, Desaware '00, and Richter '97 in combination disclose: wherein said capturing environmental information associated with computer system activity further includes capturing:

- environmental record identifier;
- time stamp indicating a time when an event described in said environmental record took effect;
- name of vendor associated with said program; and
- phone number for contacting said vendor (Knowles '96 : Table 6.4).

Note that the specification states, "Many operating systems today provide a variety of installation tools that gather much of this (environmental record) data" (Specification: p. 7, para. 0032, lns. 4-5). Since the specific fields are all information available to the operating system, Knowles '96, Desaware '00, and Richter '97 in combination reads on these fields. Additionally, note that Desaware '00 of the Knowles '96, Desaware '00, and Richter '97 combination enables providing arbitrary data to the EVENT VIEWER (TM).

Claim 22:

Regarding Claim 22, Knowles '96, Desaware '00, and Richter '97 in combination disclose all the limitations of Claim 21 (supra). Additionally, Knowles '96, Desaware '00, and Richter '97 in combination disclose: wherein said environmental flags classify a task as directed to at least one of (Richter '97: pp. 43-47, section titled, "A Process's Environmental Variables"):

- a software program;
- a hardware product;
- an initial entry;
- a change to said software program;
- a change to said hardware product;
- deletion of said software program;
- deletion of said hardware product;
- a change in said name of vendor; and
- a change in said phone number.

Note that Desaware '00 of the Knowles '96, Desaware '00, and Richter '97 combination enables providing arbitrary data to the EVENT VIEWER (TM).

Art Unit: 2171

Claim 26:

Regarding Claim 26, Knowles '96 discloses SYSTEMS MANAGEMENT SERVER (TM) (SMS), a well-known systems management product produced by MICROSOFT CORPORATION (TM). SMS (TM) uses SQL SERVER (TM), a relational database produced by MICROSOFT CORPORATION (TM), to store monitoring data. Specifically, Knowles '96 discloses: a system for managing computer performing, comprising:

- a computer system in communication with a server via a network, said computer system executing a program (Knowles '579: p. 577, section titled, "Enter Systems Management Server");
- a system monitoring application executing on said server (Knowles '579: p. 577, section titled, "Enter Systems Management Server")

Additionally, Knowles '96 discloses EVENT VIEWER (TM). Specifically, Knowles '96 additionally discloses:

- wherein for said each task executing in said computer system, said system monitoring application performs:
 - o capturing details relating to said execution while said task is active, said details including a task name assigned by a user of said server and a flag (Knowles '96: pp. 179-185 – note events read on activity records); and
 - o storing said details in an activity record (Knowles '96: pp. 179-185 – note events read on activity records).

However, Knowles '96 does not explicitly disclose:

- specifying whether said task was successfully completed;

- capturing environmental information associated with computer system activity that occurs while said task is active, said environmental information including a name of said program and environmental flags operable for classifying said task;
- creating an environmental record; and
- storing said environmental information in said environmental record.

Desaware '00 discloses the EVENT LOG TOOLKIT (TM) that enables making custom events i.e. custom flags, for applications that are visible in the EVENT VIEWER (TM) of Knowles '96 (Desaware '00: "Desaware's Event Log Toolkit (TM) for Windows 2000 (TM) and Windows NT 4.0 (TM)", pp. 1-2). However, Desaware '00 does not explicitly disclose the flag is an indication whether or not the task was successfully completed, and furthermore, Desaware '00 does not explicitly disclose environment information.

Richter '97 discloses the GetExitCodeProcess() function, which is part of the MICROSOFT WIN32 API (TM). Specifically, the GetExitCodeProcess() function of Richter '97 enables sending a status to the operating system whether or not a task was successfully completed (Richter '97: p. 69, ln. 19). Additionally, Richter '97 discloses environmental information (Richter '97: pp. 43-47, section titled, "A Process's Environmental Variables – note that this environmental information stored in an EVENT VIEWER (TM) record reads on capturing environmental data and creating and storing an environmental record).

It would have been obvious to a person having ordinary skill in the art to apply the SMS (TM) server of Knowles '96 to the EVENT VIEWER (TM) of Knowles '96. The motivation to combine is disclosed by Knowles '96 in which SMS (TM) accesses event information (Knowles '96: p. 633, Table 20.2, note access to events).

It would have been further obvious to a person having ordinary skill in the art to apply the EVENT LOG TOOLKIT (TM) of Desaware '00 to Knowles '96. The motivation to combine is on the same basis as Claim 17 (supra).

It would have been moreover obvious to a person having ordinary skill in the art to apply the Richter '97 API calls to the Knowles '96 and Desaware '00 combination. The motivation to combine is on the same basis as Claim 17 (supra).

Claims 30-32:

Regarding Claims 30-32, Knowles '96, Desaware '00, and Richter '97 in combination disclose all the limitations of Claim 26 (supra). Additionally, Knowles '96, Desaware '00, and Richter '97 in combination disclose:

- (Claim 30) wherein said details relating to said execution further include:
 - o computer system time stamp at task end;
 - o an activity record identifier;
 - o amount of initial storage allocated to said task;
 - o name of an initial program invoked for said task;
 - o a pointer to a list of additional storage requests for said task;
 - o a count of a number of additional storage requests made for said task;
 - o a pointer to a list of data sets or file names used by said program;
 - o a pointer to a list of locks and latches used by said program; and
 - o a count of a number of locks and latches used by said program (Knowles '96: p. 182, Table 6.4).

- (Claim 31) wherein said environmental information associated with said computer system activity further includes:
 - o environmental record identifier;
 - o time stamp indicating a time when an event described in said environmental record took effect;
 - o name of vendor associated with said program; and
 - o phone number for contacting said vendor (Knowles '96 : Table 6.4 – see discussion regarding Claim 21).
- (Claim 32) wherein said environmental flags classify a task as directed to at least one of:
 - o a software program;
 - o a hardware product;
 - o an initial entry;
 - o a change to said software program;
 - o a change to said hardware product;
 - o deletion of said software program deletion of said hardware product;
 - o a change in said name of vendor; and
 - o a change in said phone number (Richter '97: pp. 43-47, section titled, “A Process’s Environmental Variables” – see discussion in Claim 22).

Art Unit: 2171

3. Claims 19 and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Knowles '96, Desaware '00, and Richter '97, in view of U.S. Patent No. 4,166,290 issued to Furtman et al. (hereafter Furtman '290).

Claim 19:

Regarding Claim 19, Knowles '96, Desaware '00, and Richter '97 in combination disclose all the limitations of Claim 18 (supra). However, Knowles '96, Desaware '00, and Richter '97 in combination do not explicitly disclose: wherein if said analyzing said environmental records in conjunction with said activity records to determine a root cause of said error does not reveal a root cause of said error:

- generating a contention analysis report utilizing details provided in said activity records and said environmental records.

Furtman '290 discloses contention analysis. Specifically, Furtman '290 discloses: if said analyzing said environmental records in conjunction with said activity records to determine a root cause of said error does not reveal a root cause of said error:

- generating a contention analysis report utilizing details provided in said activity records and said environmental records (Furtman '290: col. 4, lns. 30-34).

It would have been obvious to a person having ordinary skill in the art to combine the contention analysis report of Furtman '290 with the Knowles '96, Desaware '00, and Richter '97 combination. The motivation to combine is suggested by Furtman '290, which teaches that performance monitoring is necessary for optimizing equipment (Furtman '290: col. 1, lns. 7-24). Since contention analysis indicates an area where the system configuration is sub-optimal, combining the contention analysis report of Furtman '290 with the Knowles '96, Desaware '00,

and Richter '97 combination monitor explicitly adds an advantageous report to the Knowles '96, Desaware '00, and Richter '97 combination.

Claim 29:

Regarding Claim 29, Knowles '96, Desaware '00, and Richter '97 in combination disclose all the limitations of Claim 26 (supra). However, Knowles '96, Desaware '00, and Richter '97 in combination do not explicitly disclose: further comprising a contention analysis report generated by said system analysis application; said contention analysis report created utilizing details provided in said activity records and said environmental records.

Furtman '290 discloses contention analysis. Specifically, Furtman '290 discloses: further comprising a contention analysis report generated by said system analysis application; said contention analysis report created utilizing details provided in said activity records and said environmental records (Furtman '290: col. 4, lns. 30-34).

It would have been obvious to a person having ordinary skill in the art to combine the contention analysis report of Furtman '290 with the Knowles '96, Desaware '00, and Richter '97 combination. The motivation to combine is on the same basis as Claim 19 (supra).

4. Claims 23, 27-28, and 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Knowles '96, Desaware '00, and Richter '97, in view of (U.S. Patent No. 5,835,770, issued to Shum et al. (hereafter Shum '770)).

Claim 23:

Regarding Claim 23, Knowles '96, Desaware '00, and Richter '97 in combination disclose all the limitations of Claim 17 (supra). However, Desaware '00, and Richter '97 in

combination do not explicitly disclose: executing an analysis routine; wherein said analyzing said activity records to determine a root cause of said error and said analyzing said environmental records in conjunction with said activity records to determine a root cause of said error is accomplished by said analysis routine.

Shum '770 discloses a system monitor system inquiry facility. Specifically, Shum '770 discloses: executing an analysis routine; wherein said analyzing said activity records to determine a root cause of said error and said analyzing said environmental records in conjunction with said activity records to determine a root cause of said error is accomplished by said analysis routine (Shum '770: col. 3, lns. 25-49).

It would have been obvious to a person having ordinary skill in the art to apply the system monitor of Shum '770 to the Knowles '96, Desaware '00, and Richter '97 combination. The motivation to combine

It would have been obvious to combine the ability to cumulatively collect and view activity and environment information of a task as disclosed by the Knowles '96, Desaware '00, and Richter '97 combination with the Shum '770 system inquiry facility. The motivation to combine is suggested by Knowles '96 that an application that shows cumulative errors in the past is able to show a user that a small problem is growing into a larger problem (Knowles '96: p. 72). This capability is desirable to add to the Shum '770 task status inquiry facility.

Claim 27:

Regarding Claim 27 Knowles '96, Desaware '00, and Richter '97 in combination disclose all the limitations of Claim 26 (supra). Additionally, Knowles '96, Desaware '00, and Richter '97 in combination disclose: further comprising:

- a system analysis application executing on said server; wherein upon detecting an error relating to a task, said system analysis application performs:
 - o retrieving activity records associated with said task in which said error was detected.

However, Knowles '96, Desaware '00, and Richter '97 in combination do not explicitly disclose analyzing said activity records to determine a root cause of said error.

Shum '770 discloses a system monitor system inquiry facility. Specifically, Shum '770 discloses: analyzing said activity records to determine a root cause of said error (Shum '770: col. 3, lns. 25-49 – note that the process of querying the system to determine failure events reads on analyzing activity records to determine a root cause).

It would have been obvious to combine the ability to cumulatively collect and view activity and environment information of a task as disclosed by the Knowles '96, Desaware '00, and Richter '97 combination with the Shum '770 system inquiry facility. The motivation to combine is on the same basis as Claim 23 (supra).

Claim 28:

Regarding Claim 28, Knowles '96, Desaware '00, Richter '97, and Shum '770 in combination discloses all the limitations of Claim 27 (supra). Additionally, Knowles '96, Desaware '00, Richter '97, and Shum '770 in combination disclose: wherein if said analyzing said activity records does not reveal a root cause of said error, said system analysis application further performs:

- retrieving environmental records created between a last successful execution of a program associated with said task and a time in which said error occurred (Shum

'770: col. 3, lns. 25-49 – note inquiry facility; Knowles '96: pp. 181-182 – note section titled, “Viewing Events”); and

- analyzing said environmental records in conjunction with said activity records to determine a root cause of said error (Knowles '96: p.182, Table 6.4; Richter '97: pp. 43-47, section titled, “A Process’s Environmental Variables”).

Claim 33:

Regarding Claim 33 Knowles '96, Desaware '00, and Richter '97 in combination disclose all the limitations of Claim 26 (supra). However, Knowles '96, Desaware '00, and Richter '97 in combination do not explicitly disclose: further comprising: an analysis routine executing on said server; wherein said analyzing said activity records to determine a root cause of said error and said analyzing said environmental records in conjunction with said activity records to determine a root cause of said error is accomplished by said analysis routine.

Shum '770 discloses a system monitor system inquiry facility. Specifically, Shum '770 discloses: further comprising: an analysis routine executing on said server; wherein said analyzing said activity records to determine a root cause of said error and said analyzing said environmental records in conjunction with said activity records to determine a root cause of said error is accomplished by said analysis routine (Shum '770: col. 3, lns. 25-49 – note the inquiry facility reads on an analysis routine).

It would have been obvious to combine the ability to cumulatively collect and view activity and environment information of a task as disclosed by the Knowles '96, Desaware '00, and Richter '97 combination with the Shum '770 system inquiry facility. The motivation to combine is on the same basis as Claim 23 (supra).

Response to Arguments

5. Applicant's arguments filed April 27, 2004 have been fully considered but they are not persuasive. Specifically: Examiner notes that Applicant has amended claims to reflect the specific details of the activity record and the environmental record as follows:

New Limitations of Claims 17 and 24:

- capturing details for every task while each task is active;
- storing details in an activity record;
- retrieve activity records;
- assigning a task name and successful completion flag (Amendment: p. 11, lns. 11-16);

New Limitations of Claims 18 and 25:

- capturing environmental information;
- retrieving environmental information;
- environmental flags (Amendment: p. 15, lns. 1-19).

Examiner reiterates argument from prior office action that both the EVENT VIEWER (TM) of Knowles '96 and the SMS (TM) of Knowles '96 monitor task details. An event log record in the EVENT VIEWER (TM) corresponds to an activity record. Furthermore, the EVENT VIEWER (TM) is designed to be arbitrarily extensible. Examiner provides the Desaware '00 reference as one of many means to extend the EVENT VIEWER (TM). In fact,

Art Unit: 2171

Examiner observes that Desaware '00 merely wraps the preexisting EVENT LOG API (TM) provided by MICROSOFT CORPORATION (TM) for the EVENT VIEWER (TM). Thus any information, provided through the EVENT LOG API (TM) enabled by Desaware '00 is accessible to the EVENT VIEWER (TM). The environmental and flag information specified by applicant is readily available from the operating system, as disclosed by Richter '97. Combined with Knowles '96 and Desaware '00, this indicates that environmental information is accessible via the EVENT VIEWER. For the networked embodiment, Examiner observes SMS (TM) of Knowles '96 has access to EVENT VIEWER events. For the formal presentation of this argument, see 103(a) rejections above regarding Claims 17-18 and 24-25.

In general, tasks (e.g. processes, threads, jobs, and the like), are operating system kernel objects, and therefore all information about a task is readily accessible via the operating system APIs (e.g. Richter '97). Combined with a generic system monitor (e.g. Knowles '96), logging any activity and environment information is enabled.

6. Examiner responds to Applicant's individual arguments:

Argument 1 (regarding Shum '770 with respect to Claims 17 and 24):

Argument 1a: Applicant asserts, that the "storage blocks" of Shum '770 do not read on

Applicant's "activity records that relate to the execution of the task".

Response 1a: The combination of Shum '770 and the activity records of Knowles '96 reads on storing activity records. Shum '770 may disclose "storage blocks", but combined with Knowles '96, the data is explicitly formatted into "activity records."

In general, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

Argument 1b: Applicant asserts, that the "status inquiry command" of Shum '770 does not read on the Applicant's "task name."

Response 1b: On the contrary, Examiner observes that Shum '770 is able to select an arbitrary task from which to obtain information (Shum '770: col. 6, lns. 3-24).

Additionally, Examiner recalls Applicant's statement, "Thus, without a task name and without an activity record generated, for 'each execution of a task', Shum '770 would not be capable of retrieving its stored information in the manner as recited in the instant application." Furthermore, Shum '770 and Knowles '96 in combination, provides activity records in the manner of the instant application. Since Shum '770 is in fact able to select an arbitrary task, the "status inquiry command" thus contains task identification/task naming information.

Argument 1c: Applicant asserts, that Shum '770 does not recite "analyzing the activity records to determine a root cause of error."

Response 1c: On the contrary. Examiner observes that using the inquiry facility of Shum '770 (Shum '770: col. 3, lns. 25-49) to determine a cause of failure reads upon "analyzing the activity records to determine a root cause of error." See discussion regarding Claim 23 (supra).

Argument 2 (regarding Knowles '96 with respect to Claims 17 and 24):

Argument 2: Applicant asserts Knowles '96 does not recite specific features of Claims 17 and 24, but rather only "limited operating system techniques."

Response 2: On the contrary, Examiner observes that operating system techniques are not limited to operations on applications such as the EVENT VIEWER (TM), but also programmatically. Examiner observes that any application functionality to monitor processes, threads, tasks, jobs, and any other Operating System Kernel object must in fact be enabled by the operating system. See discussion in paragraph 5 (supra) in which Knowles' 96 combined with Desaware '00 (an extension mechanism) and Richter '97 (WINDOW 32 API (TM)), enables both activity and environmental information to be accessible via the EVENT VIEWER (TM).

Argument 3 (regarding Fujii '898 with respect to Claims 17 and 24):

Argument 3: Applicant asserts the “error handling operation” of Fujii '898 is not in response to the “activity records” as disclosed by Applicant.

Response 3: Present Office Action does not rely on Fujii '898, thus arguments regarding Fujii '898 are moot.

Argument 4 (regarding Furtman '290 with respect to Claims 19 and 29):

Argument 4: Applicant asserts that the “contention analysis” of Furtman '290 is not in response to the “activity records” as disclosed by Applicant.

Response 4: Furtman '290 in combination with Knowles '96 has Furtman '290 reading and analyzing records in Knowles '96. Thus, the combination of Furtman '290 and the activity records of Knowles '96 reads on Furtman '290 responding to activity records. Additionally, see *In re Keller* discussion in Response 1a (supra).

Argument 5 (regarding Furtman '290 with respect to Claims 18-25):

Argument 5: Applicant asserts that the “contention analysis” of Furtman '290 is not in response to the “environmental records” as disclosed by Applicant.

Response 5: Furtman '290 in combination with Knowles '96, Desaware '00, and Richter '97 has Furtman '290 reading and analyzing records in Knowles '96, Desaware '00, and Richter '97 in combination. Thus, the combination of Furtman '290 and the environment records of Knowles '96, Desaware '00, and Richter '97 reads on Furtman '290 responding to environment records. See rejection of Claims 19 and 29 (*supra*). Additionally, see *In re Keller* discussion in Response 1a (*supra*).

Conclusion

7. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).


A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Patrick J Santos whose telephone number is 703-305-0707. The examiner can normally be reached on M-F 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Safet Metjahic can be reached on 703-308-1436. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Patrick J.D. Santos
July 2, 2004


SAFET METJAHIC
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100